

1) Conceptos clave. Define:

- XML
- Metalenguaje
- Analizador sintáctico
- Parser
- Handler
- DOM
- SAX
- JAXB/Jakarta
- Binding
- Navegación bidireccional en XML

2) Comparativa DOM, SAX, JAXB

- Ventajas e inconvenientes de cada uno (*comparar memoria/velocidad/edición*)
- Explica las clases e instrucciones (métodos) más utilizadas en cada modelo:

DOM

- DocumentBuilderFactory.newInstance()
- DocumentBuilder.parse(...)
- Document.getDocumentElement()
- getElementsByTagName(...)
- Element.getAttribute(...) / setAttribute(...)
- Node.getTextContent() / setTextContent(...)
- Document.createElement(...), appendChild(...), removeChild(...)
- TransformerFactory.newInstance()

SAX

- SAXParserFactory.newInstance(), setNamespaceAware(...)
- SAXParser.parse(File|InputSource, handler)
- DefaultHandler:
 - o startElement(...) / endElement(...)
 - o characters(char[], start, length)

JAXB

- JAXBContext.newInstance(ClaseRaíz.class)
- Unmarshaller /Marshaller
- Anotaciones:
 - o @XmlRootElement, @XmlAccessorType(XmlAccessType.FIELD)
 - o @XmlElement, @XmlAttribute, @XmlElementWrapper
 - o @XmlTransient

3) Resuelve

1. V/F: DOM carga todo el XML en memoria. Justifica la respuesta.
2. Si tuvieras que procesar un XML de 5 GB, ¿qué enfoque usarías?
a) DOM b) SAX c) JAXB
3. ¿Qué añadir en Java 17 para usar JAXB?
4. Diferencia entre bien formado y válido en XML.
5. Justifica qué usarías y por qué (criterios de memoria, rendimiento, mantenibilidad, validación):
 - a) Procesar un XML de 2 GB de logs para extraer estadísticas.
 - b) Editar y guardar un XML de configuración pequeño que cambia a menudo.